

## Controllo di Arduino con uno smartphone Android tramite Bluetooth

### Obiettivo del tutorial

Il Bluetooth, grazie al basso consumo energetico e al costo molto contenuto, è una delle tecnologie di comunicazione wireless più diffuse. Il tutorial illustra come utilizzare Bluetooth per far comunicare una scheda Arduino con uno smartphone Android.

### Il progetto

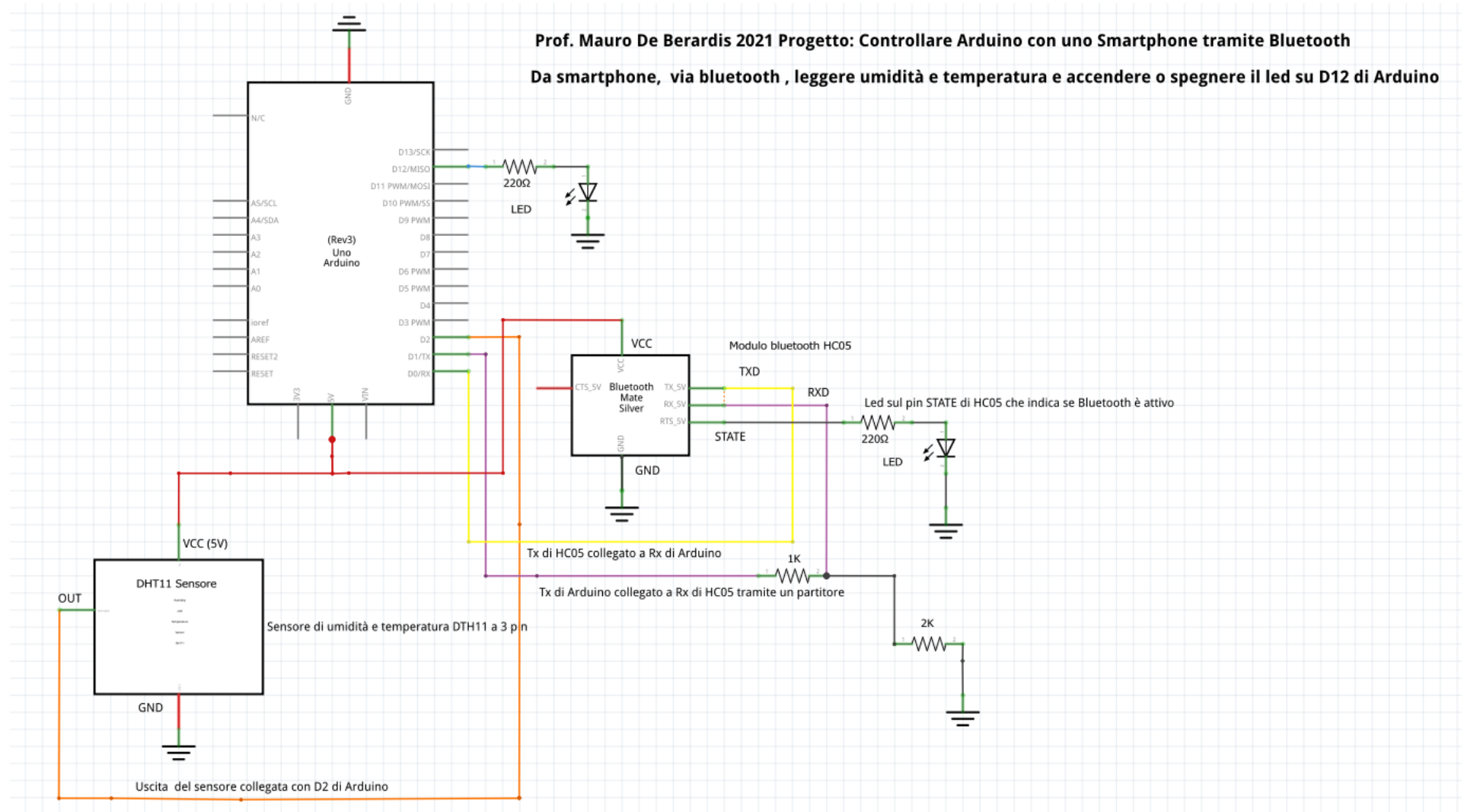
Uno smartphone Android interfacciandosi con Arduino tramite Bluetooth:

- a) legge i valori di temperatura ed umidità misurati da un sensore
- b) pilota l'accensione e lo spegnimento di un led (simulazione dell'accensione/spegnimento di un motore o di una caldaia, di un impianto di irrigazione o di un cancello automatico, delle luci di un giardino o di una telecamera di sorveglianza, ecc.)

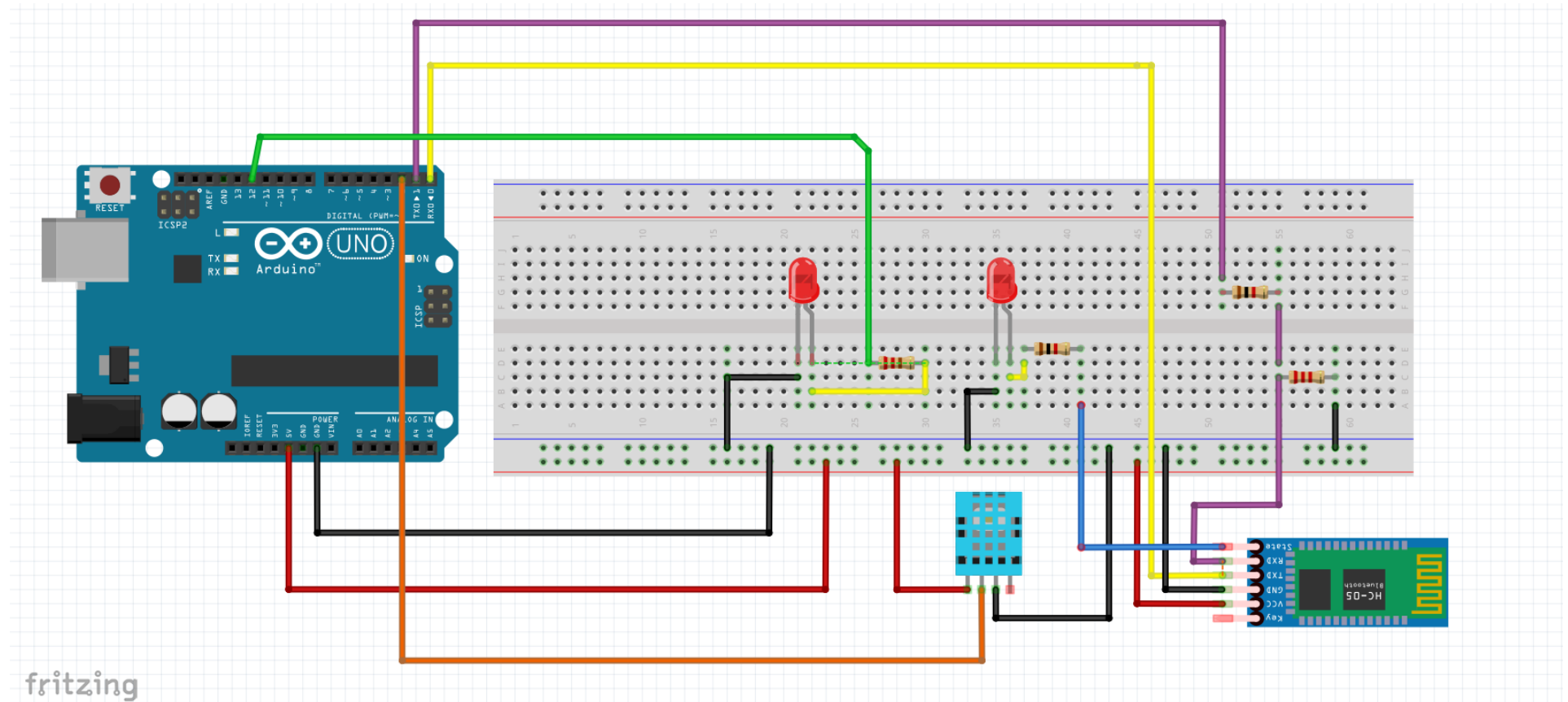
### Che cosa serve

- pc Windows X con Ide Arduino e Android Studio
- smartphone Android con Bluetooth
- scheda Arduino UNO collegata al pc con cavo USB
- breadboard con cavetteria
- **modulo Bluetooth HC05**
- **sensore di umidità e temperatura DHT11**
- 2 led
- 2 resistenze 220  $\Omega$
- 3 resistenze 1K $\Omega$

## Schema elettrico



## Collegamenti sulla breadboard



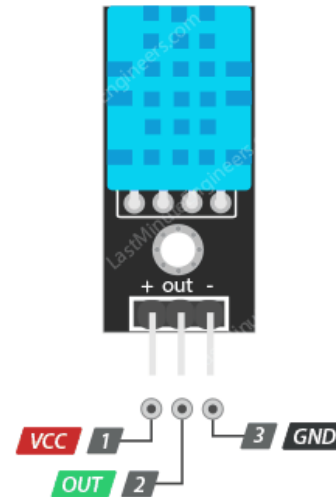
## Connessioni tra modulo Bluetooth HC05 e Arduino

Modulo Bluetooth HC05	Arduino
STATE	-
TXD	RX (pin 0)
RXD	TX (pin 1)
GND	5V
VCC	GND
EN	-



## Connessioni tra sensore di umidità e temperatura DHT11 e Arduino

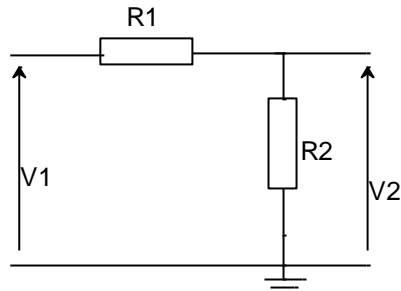
Sensore DHT11	Arduino
VCC	5V
DATA (I/O)	D2
GND	GND



## Chiarimenti

- Il modulo Bluetooth HC05 può essere alimentato da 3,6 a 6 volt perché contiene un regolatore di tensione
- Il pin STATE del modulo Bluetooth viene monitorato tramite un led. Quando il led è acceso vuol dire che il modulo HC05 è connesso al Bluetooth dello smartphone

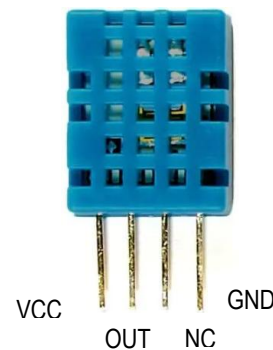
- Poichè il livello di tensione logica dei pin dati del modulo Bluetooth HC05 è 3,3 V e il pin TXD di Arduino ha un'uscita a 5V, per evitare il rischio di bruciare il modulo HC05, TXD di Arduino ed RX di HC05 vengono collegati tramite un partitore di tensione. Per il circuito di prova ho utilizzato una resistenza R1=1000  $\Omega$  e una resistenza R2=2000  $\Omega$  (per realizzare R2 ho messo in serie 2 resistenze di 1000  $\Omega$ )



$$V2 = R2 \times I \quad I = \frac{V1}{R1 + R2} \quad \text{quindi } V2 = \frac{R2 \times V1}{R1 + R2}$$

$$\text{Nel nostro caso: } V2 = \frac{2000 \times 5}{3000} = 3.3 \text{ Volt}$$

- Il pin TXD del modulo Bluetooth e il pin RX di Arduino possono essere collegati direttamente perché il segnale 3.3V proveniente dal pin Tx del modulo Bluetooth è sufficiente per essere accettato come valore logico alto sulla scheda Arduino
- Il DHT11 è un sensore digitale di umidità e temperatura dell'aria costituito da una parte resistiva che si occupa della rilevazione dell'umidità e da un NTC che rileva la temperatura. Il sensore viene fabbricato in due configurazioni: a 3 o a 4 pin. In entrambi i casi i pin VCC, GND e OUT (DATA), che devono essere collegati ad Arduino, sono indicati chiaramente. Il pin NC, nel caso della configurazione a 4 pin, non viene collegato. Tra la linea del segnale OUT e VCC (5V) è necessaria una resistenza di pull-up da 10K Ohm. Nel caso di configurazione a 3 pin, utilizzata in questo progetto ed assemblata su un circuito stampato (PCB), la resistenza di pull-up è già montata.



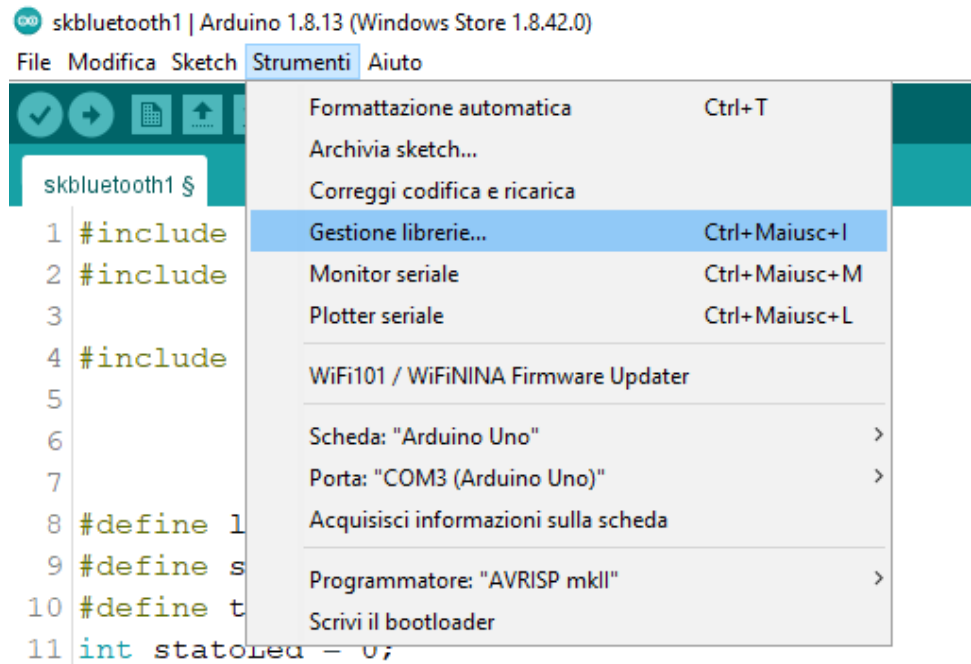
## Sketch di Arduino

### Importante!!!

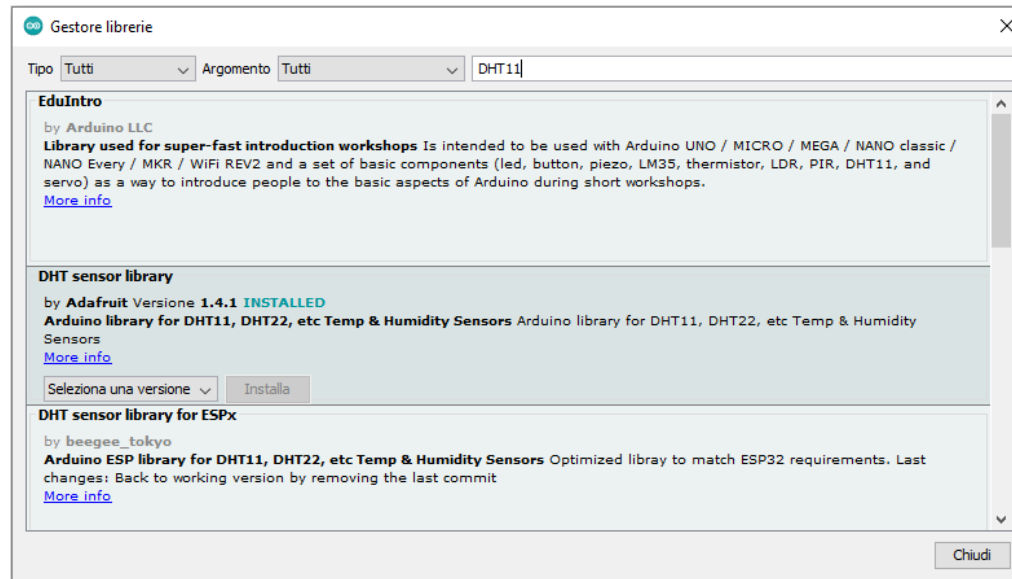
Ogni volta che si carica lo sketch su Arduino, consiglio di togliere l'alimentazione al modulo bluetooth scollegando il pin VCC del modulo HC05.

Prima di iniziare la programmazione di Arduino, bisogna installare, e quindi includere nello sketch, due librerie necessarie per poter utilizzare il sensore di umidità e temperatura DHT11.

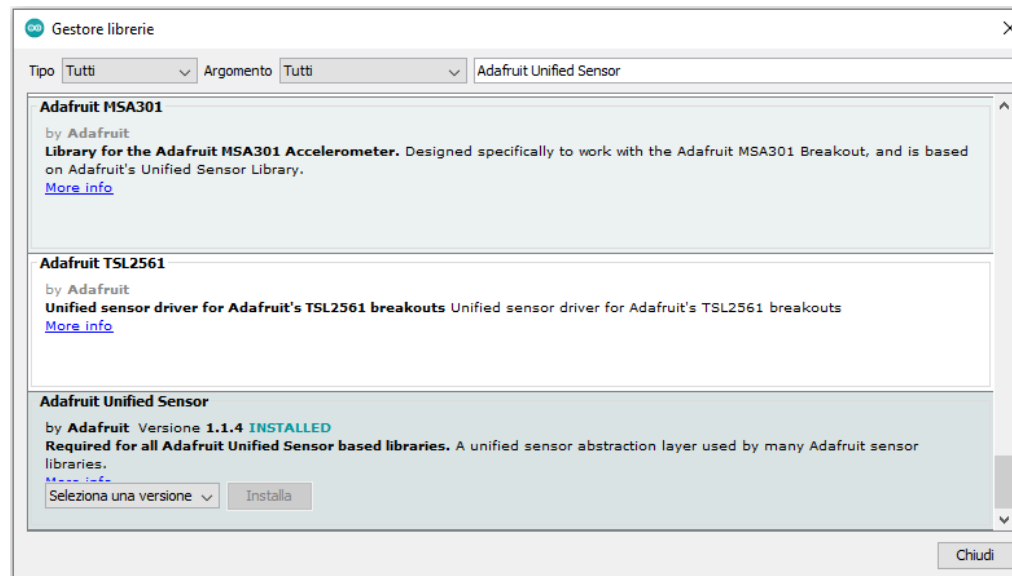
- Con l'IDE Arduino in esecuzione, scegliere *Strumenti-Gestione librerie...*



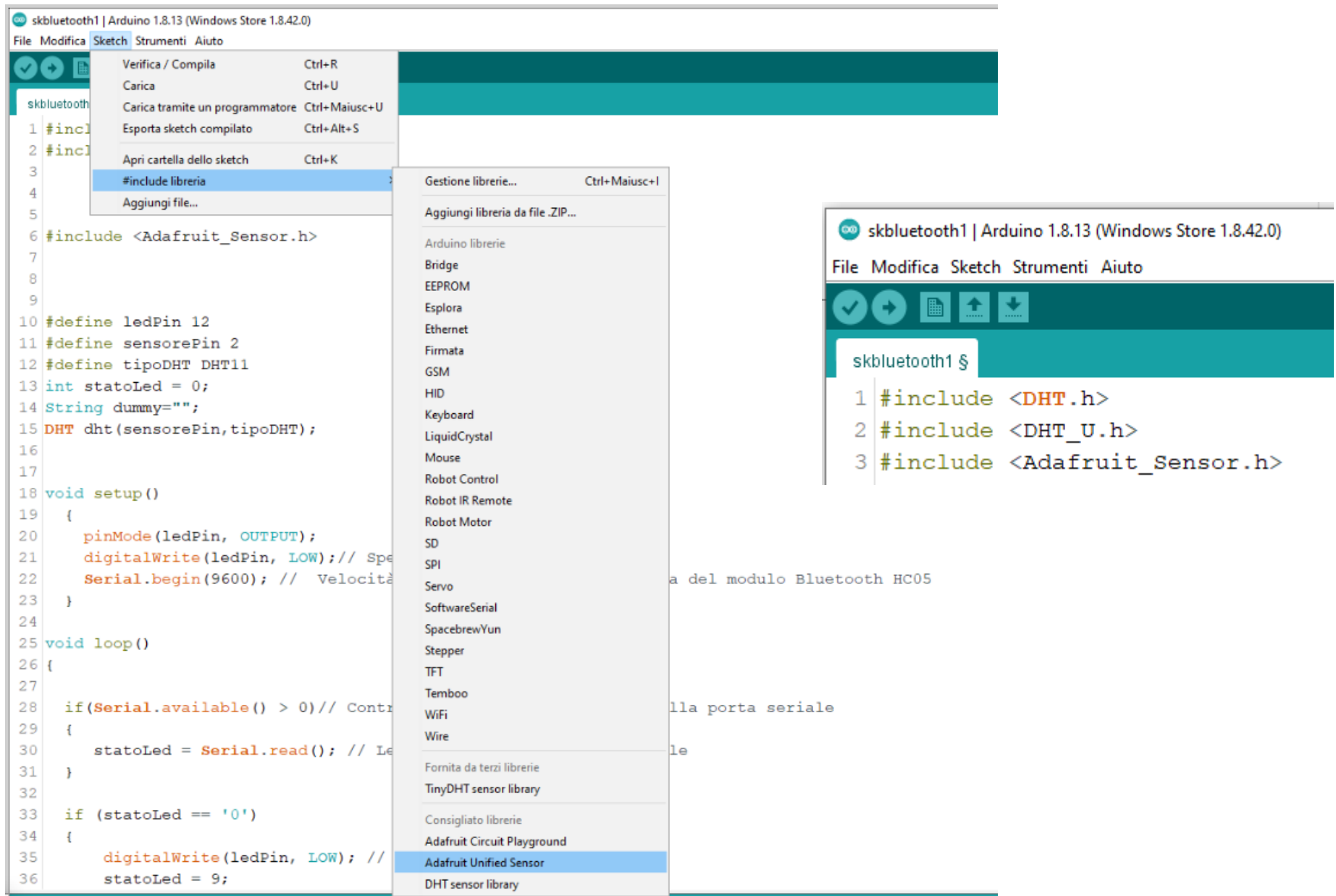
- Quando si apre la finestra del Gestore librerie, nel campo di ricerca inserire "*DHT11*"
- Tra le diverse librerie proposte trovare, selezionare e installare "**DHT sensor library**" ( by Adafruit)



- Subito dopo, allo stesso modo inserire nel campo di ricerca "*Adafruit Unified Sensor*".
- Tra le diverse librerie proposte trovare, selezionare e installare "**Adafruit Unified Sensor**" ( by Adafruit)



- Una volta installate le due librerie, aprire lo sketch che si vuole programmare, scegliere l'opzione *Sketch-#include libreria* e selezionare prima "DHT Sensor Library" e poi "Adafruit Unified Sensor" (le due librerie vengono incluse nello sketch)





kbluetooth1.ino

```
#include <DHT.h>
#include <DHT_U.h>
#include <Adafruit_Sensor.h>
#define ledPin 12      // pin di Arduino collegato al led che vogliamo comandare da smartphone
#define sensorePin 2   // pin di Arduino collegato all'uscita (pin OUT) del sensore DHT11
#define tipoDHT DHT11 // definisce il tipo di sensore della famiglia DHT: in questo caso viene selezionato DHT11
                        // ma esistono sono altri sensori quali DHT21 e DHT22

int statoLed = 0;
String dummy="";
DHT dht(sensorePin,tipoDHT);// inizializzazione del sensore DHT11

void setup()
{
  pinMode(ledPin, OUTPUT);
  digitalWrite(ledPin, LOW); // Spegne il led di default
  Serial.begin(9600); // Velocità di comunicazione predefinita del modulo Bluetooth HC05
}

void loop()
{
  if(Serial.available() > 0)
  {
    statoLed = Serial.read(); // Legge i dati dalla porta seriale
    // lo smartphone invia 1 per accendere il led e 0 per spegnerlo
    // invia 2 per leggere l'umidità e la temperatura misurate dal sensore DHT11
    // quindi sulla porta seriale, se ci sono dati, Arduino può leggere esclusivamente 0 oppure 1 oppure 2
  }
  if (statoLed == '0')
  {
    digitalWrite(ledPin, LOW); // Spegne il led
    statoLed = 9;
  }
}
```

```

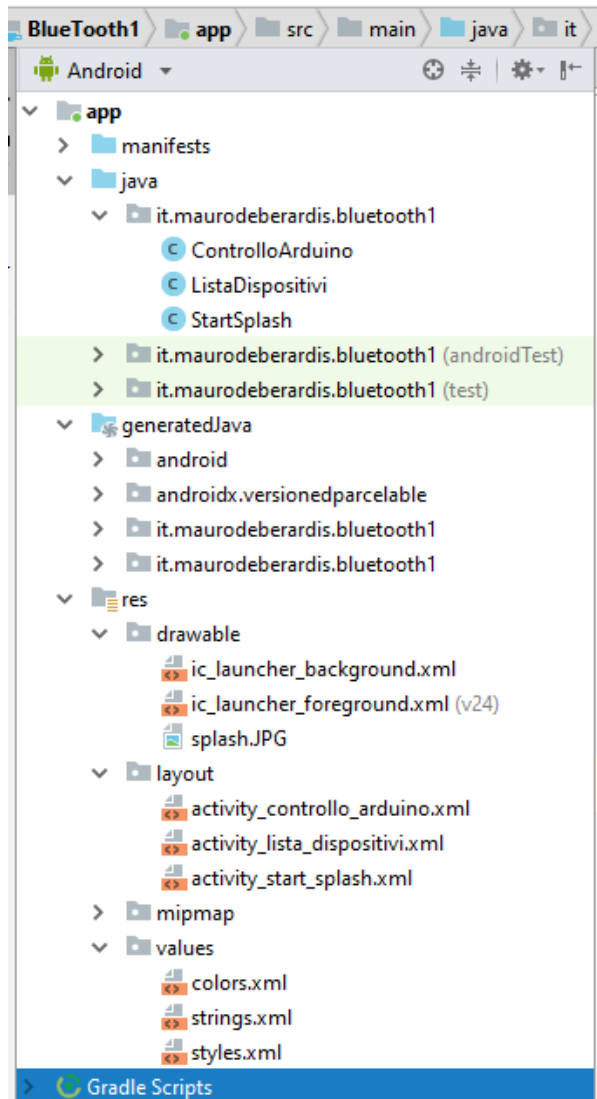
if (statoLed == '1')
{
    digitalWrite(ledPin, HIGH); // Accende il led
    statoLed = 9;
}
if (statoLed == '2')
{
    dht.begin();
    float h,t;
    // faccio 50 letture in modo che i valori letti dal sensore, che non è velocissimo, si stabilizzino
    for (int i = 0; i < 50; i++)
    {
        delay(25);
        // pulisce la seriale in maniera che alla fine delle 50 letture ci sia una sola coppia di valori
        // da trasmettere allo smartphone
        while (Serial.available() > 0) dummy=Serial.read();
        // Legge i valori del sensore
        h = dht.readHumidity(); // Lettura dell'umidità
        t = dht.readTemperature(); // Lettura della temperatura in gradi Celsius
    }
    // ultima delle 50 letture
    if (isnan(h) || isnan(t)) //Verifica se si presenta un errore di lettura
    {
        Serial.println("errore,errore");
    }
    else
    {
        String umidita=String(h);
        String temperatura=String(t);
        String s=umidita+","+temperatura;
        Serial.print(s); // scrive i dati sulla porta seriale
        statoLed = 9;
    }
}

} // chiude if (statoLed == '2')-----

} // chiude void loop()-----

```

## App Android



### AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="it.maurodeberardis.bluetooth1">
    <uses-permission android:name="android.permission.BLUETOOTH_ADMIN" />
    <uses-permission android:name="android.permission.BLUETOOTH" />
    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/AppTheme">
        <activity android:name=".StartSplash">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <activity android:name=".ListaDispositivi" />
        <activity android:name=".ControlloArduino"></activity>
    </application>
</manifest>
```

### strings.xml

```
<resources>
    <string name="app_name">Controllo di Arduino via Bluetooth</string>
</resources>
```

**activity\_start\_splash.xml**

```

<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".StartSplash"
    tools:layout_editor_absoluteY="81dp">
    <TextView
        android:id="@+id/textView2"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginStart="8dp"
        android:layout_marginLeft="8dp"
        android:layout_marginTop="64dp"
        android:layout_marginEnd="8dp"
        android:layout_marginRight="8dp"
        android:gravity="center"
        android:paddingLeft="10dp"
        android:paddingRight="10dp"
        android:text="Per eseguire l'app devi essere connesso al Bluetooth di Arduino"
        android:textColor="#000080"
        android:textSize="18dp"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/imgsplash" />
    <TextView
        android:id="@+id/textView3"
        android:layout_width="wrap_content"
        android:layout_height="15dp"
        android:layout_marginBottom="16dp"
        android:text="Mauro De Berardis 2021"
        android:textColor="#000080"
        app:layout_constraintBottom_toBottomOf="parent"

```

```

app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="0.506"
app:layout_constraintLeft_toLeftOf="parent"
app:layout_constraintRight_toRightOf="parent"
app:layout_constraintStart_toStartOf="parent" />
<TextView
    android:id="@+id/textView1"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginStart="8dp"
    android:layout_marginLeft="8dp"
    android:layout_marginTop="24dp"
    android:layout_marginEnd="8dp"
    android:layout_marginRight="8dp"
    android:gravity="center"
    android:text="App per controllare Arduino tramite Bluetooth"
    android:textColor="#000080"
    android:textSize="18dp"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintRight_toRightOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />
<ImageView
    android:id="@+id/imgsplash"
    android:layout_width="wrap_content"
    android:layout_height="113dp"
    android:layout_marginStart="8dp"
    android:layout_marginLeft="8dp"
    android:layout_marginTop="100dp"
    android:layout_marginEnd="8dp"
    android:layout_marginRight="8dp"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/textView1"
    app:srcCompat="@drawable/splash" />
</android.support.constraint.ConstraintLayout>

```



**StartSplash.java**

```

package it.maurodeberardis.bluetooth1;

import android.content.Intent;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
public class StartSplash extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_start_splash);
        Thread timerThread = new Thread(){
            public void run(){
                try{
                    sleep(1500);
                }catch (InterruptedException e){
                    e.printStackTrace();
                }finally{
                    Intent intent = new Intent(StartSplash.this,ListaDispositivi.class);
                    startActivity(intent);
                }
            }
        };
        timerThread.start();
    }
    @Override
    protected void onPause() {
        super.onPause();
        finish();
    }
}

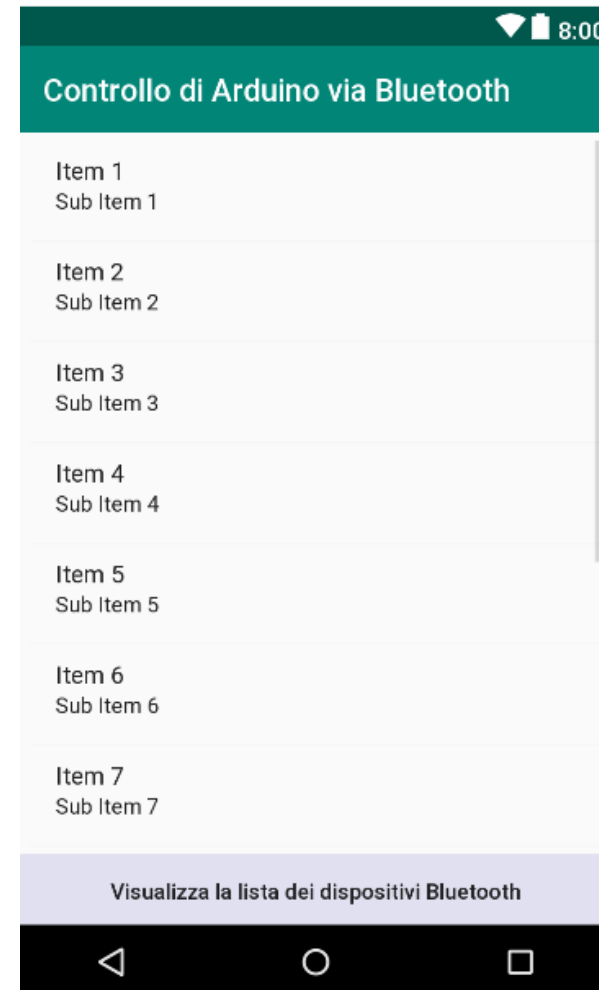
```

**activity\_lista\_dispositivi.xml**

```

<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".ListaDispositivi">
    <ListView
        android:id="@+id/listadispositivi"
        android:layout_width="368dp"
        android:layout_height="460dp"
        android:layout_marginStart="8dp"
        android:layout_marginLeft="8dp"
        android:layout_marginTop="8dp"
        android:layout_marginEnd="8dp"
        android:layout_marginRight="8dp"
        app:layout_constraintBottom_toTopOf="@+id/bLista"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.0"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />
    <Button
        android:id="@+id/bLista"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:textAllCaps="false"
        android:background="#E0E0F0"
        android:text="Visualizza la lista dei dispositivi Bluetooth"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.0"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/listadispositivi" />
</android.support.constraint.ConstraintLayout>

```



**ListaDispositivi.java**

```

package it.maurodeberardis.bluetooth1;

import android.app.ProgressDialog;
import android.bluetooth.BluetoothAdapter;
import android.bluetooth.BluetoothDevice;
import android.content.Intent;
import android.graphics.Color;
import android.os.AsyncTask;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.ListView;
import android.widget.TextView;
import android.widget.Toast;

import java.util.ArrayList;
import java.util.Set;

public class ListaDispositivi extends AppCompatActivity {
    ListView listadispositivi;
    Button btnlista;
    private BluetoothAdapter BTA;
    private Set<BluetoothDevice> DispositiviAccoppiati;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_lista_dispositivi);
        listadispositivi=findViewById(R.id.listadispositivi);
        btnlista=findViewById(R.id.btnlista);
        ControllaBluetooth();//Controlla se bluetooth è attivato
    }
}

```



```

btnlista.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {

        LeggiListaDispositivi();
    }
});
listadispositivi.setOnItemClickListener(new AdapterView.OnItemClickListener() {
    @Override
    public void onItemClick(AdapterView<?> parent, View view, int position, long id) {
        String s=((TextView)view).getText().toString();
        //s è la stringa con nome e indirizzo del dispositivo bluetooth selezionato
        Intent i = new Intent(getApplicationContext(), ControlloArduino.class);
        //richiama l'attività ControlloArduino e trasmette la stringa s
        i.putExtra("info_disp_bt",s);
        startActivity(i);    finish();
    }
});
} //chiude onCreate-----

@Override
public void onBackPressed() {
    finish();
    System.exit(0);
}

private void ControllaBluetooth()
{
    BTA = BluetoothAdapter.getDefaultAdapter();
    if(!BTA.isEnabled())
    {
        Intent i = new Intent(BluetoothAdapter.ACTION_REQUEST_ENABLE);
        startActivityForResult(i,1);
        //attiva la finestra di dialogo che invita l'utente ad attivare il Bluetooth
    }
}

```

```

    if(BTA== null)
    {
        Toast.makeText(getApplicationContext(), "Bluetooth non disponibile!", Toast.LENGTH_LONG).show();
        ControllaBluetooth();
    }
}
private void LeggiListaDispositivi()
{
    DispositiviAccoppiati = BTA.getBondedDevices();
    ArrayList lista = new ArrayList();
    if (DispositiviAccoppiati.size()>0)
    {
        for(BluetoothDevice BTD :DispositiviAccoppiati)
        {
            lista.add(BTD.getName() + "\n" + BTD.getAddress());
            //visualizza nome e mac-address di ciascun dispositivo accoppiato
        }
    }
    else
    {
        String msg="Non ci sono dispositivi Bluetooth accoppiati o Bluetooth non è attivo";
        Toast.makeText(getApplicationContext(),msg , Toast.LENGTH_LONG).show();
    }
    final ArrayAdapter adapter = new ArrayAdapter(this,android.R.layout.simple_list_item_1, lista);
    listadispositivi.setAdapter(adapter);
}
}
} //chiude class ListaDispositivi-----

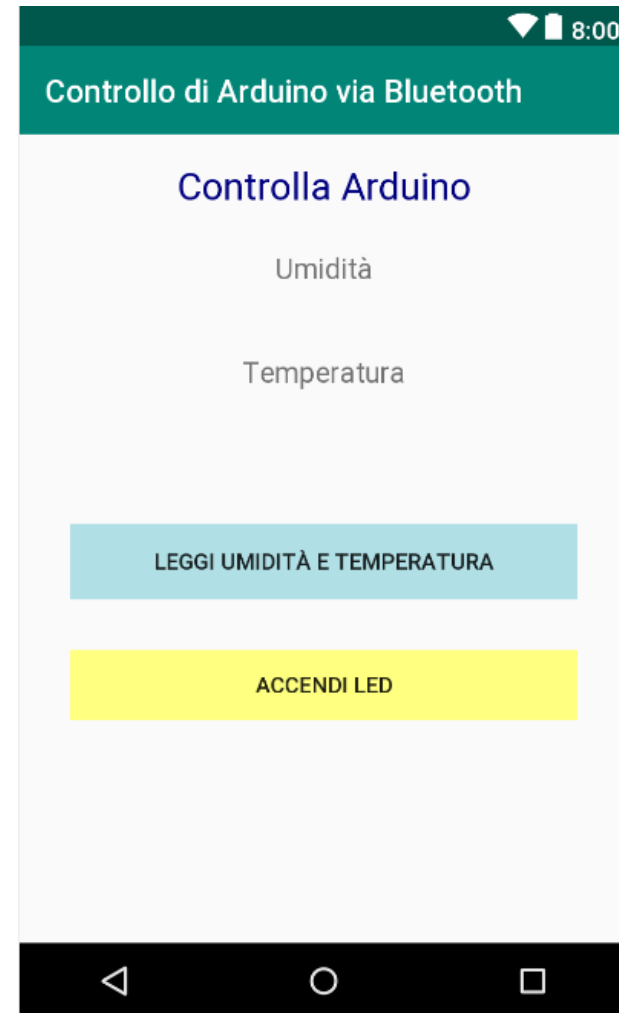
```

**activity\_controllo\_arduino.xml**

```

<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".ControlloArduino">
    <TextView
        android:id="@+id/textView"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginStart="8dp"
        android:layout_marginLeft="8dp"
        android:layout_marginTop="16dp"
        android:layout_marginEnd="8dp"
        android:layout_marginRight="8dp"
        android:gravity="center"
        android:text="Controlla Arduino"
        android:textColor="#000080"
        android:textSize="24dp"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />
    <TextView
        android:id="@+id/textView5"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginStart="8dp"
        android:layout_marginLeft="8dp"
        android:layout_marginTop="24dp"
        android:layout_marginEnd="8dp"
        android:layout_marginRight="8dp"
        android:gravity="center"
        android:text="Umidità"

```



```

        android:textSize="18dp"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.0"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/textView" />
<TextView
    android:id="@+id/tUmidita"
    android:layout_width="match_parent"
    android:layout_height="29dp"
    android:layout_marginStart="8dp"
    android:layout_marginLeft="8dp"
    android:layout_marginTop="4dp"
    android:layout_marginEnd="8dp"
    android:layout_marginRight="8dp"
    android:gravity="center"
    android:text=""
    android:textSize="24sp"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.0"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/textView5" />
<TextView
    android:id="@+id/textView6"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginStart="8dp"
    android:layout_marginLeft="8dp"
    android:layout_marginTop="8dp"
    android:layout_marginEnd="8dp"
    android:layout_marginRight="8dp"
    android:gravity="center"
    android:text="Temperatura"
    android:textSize="18dp"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.0"

```

```

app:layout_constraintLeft_toLeftOf="parent"
app:layout_constraintRight_toRightOf="parent"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toBottomOf="@+id/tUmidita" />
<TextView
    android:id="@+id/tTemperatura"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginStart="8dp"
    android:layout_marginLeft="8dp"
    android:layout_marginTop="4dp"
    android:layout_marginEnd="8dp"
    android:layout_marginRight="8dp"
    android:gravity="center"
    android:text=""
    android:textSize="24sp"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.0"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/textView6" />
<Button
    android:id="@+id/bLeggi"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginStart="32dp"
    android:layout_marginLeft="32dp"
    android:layout_marginTop="48dp"
    android:layout_marginEnd="32dp"
    android:layout_marginRight="32dp"
    android:background="#B0E0E6"
    android:text="Leggi Umidità e Temperatura"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/tTemperatura" />
<Button
    android:id="@+id/bLed"
    android:layout_width="match_parent"

```

```
android:layout_height="44dp"
android:layout_marginStart="32dp"
android:layout_marginLeft="32dp"
android:layout_marginTop="32dp"
android:layout_marginEnd="32dp"
android:layout_marginRight="32dp"
android:background="#ffff80"
android:text="Accendi Led"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="0.0"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toBottomOf="@+id/bLeggi" />
```

```
</android.support.constraint.ConstraintLayout>
```

**ControlloArduino.java**

```

package it.maurodeberardis.bluetooth1;

import android.bluetooth.BluetoothAdapter;
import android.bluetooth.BluetoothDevice;
import android.bluetooth.BluetoothSocket;
import android.content.DialogInterface;
import android.content.Intent;
import android.graphics.Color;
import android.os.AsyncTask;
import android.support.v7.app.AlertDialog;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;
import android.widget.Toast;
import java.io.IOException;
import java.util.Timer;
import java.util.TimerTask;
import java.util.UUID;

public class ControlloArduino extends AppCompatActivity {
    String info_dispositivo_bt, indirizzo=null;
    Button bLeggi,bLed;
    TextView umidita,temperatura;
    Boolean LedOn=false;
    BluetoothAdapter BTAdapter=null;
    BluetoothSocket BTSocket=null;
    BluetoothDevice BTdevice;
    private boolean BTConnesso = false;
    //In Bluetooth, tutti gli oggetti sono identificati da UUID.
    // Questi includono servizi, caratteristiche e molte altre cose.
    // Qui si utilizza un UUID standard per HC05
    static final UUID U = UUID.fromString("00001101-0000-1000-8000-00805F9B34FB");

```

```

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_controllo_arduino);
    Intent i = getIntent();
    info_dispositivo_bt = i.getStringExtra("info_disp_bt");
    //leggo le informazioni sul dispositivo bluetooth provenienti dall'attività
    // ListaDispositivi
    indirizzo=info_dispositivo_bt.substring(info_dispositivo_bt.length() - 17);
    // il mac address del dispositivo è dato dagli ultimi 17 caratteri della stringa ricevuta
    // Toast.makeText(getApplicationContext(), indirizzo, Toast.LENGTH_LONG).show();
    bLeggi=findViewById(R.id.bLeggi);
    bLed=findViewById(R.id.bLed);
    umidita=(TextView)findViewById(R.id.tUmidita);
    temperatura=(TextView)findViewById(R.id.tTemperatura);
    Connetti conn=new Connetti();
    conn.execute(); //connessione a bluetooth
    bLed.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            if(!BTConnesso){
                displaymsg();
                return;
            }
            try
            {
                if(LedOn)
                {
                    BTSocket.getOutputStream().write("0".getBytes());
                    LedOn=false;
                    bLed.setBackgroundColor(Color.parseColor("#FFFF80"));
                    bLed.setText("Accendi Led");
                }
                else
                {
                    BTSocket.getOutputStream().write("1".getBytes());
                    LedOn=true;
                }
            }
            catch (IOException e) {
                e.printStackTrace();
            }
        }
    });
}

```



```

        bLed.setBackgroundColor(Color.parseColor("#FFA070"));
        bLed.setText("Spegni Led");
    }
}
catch (IOException e)
{
    Toast.makeText(getApplicationContext(), e.toString(), Toast.LENGTH_LONG).show();
}
}
});

bLeggi.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        bLeggi.setEnabled(false); // disabilito il bottone Leggi...
        bLed.setEnabled(false); // disabilito il bottone che accend/spegne il led...
        Timer timer1=new Timer();
        timer1.schedule(new TimerTask() {
            @Override
            public void run() {
                runOnUiThread(new Runnable() {

                    @Override
                    public void run() {
                        bLeggi.setEnabled(true);
                        bLed.setEnabled(true);
                    }
                });
            }
        }, 3000);
        // ri-abilito i bottoni Leggi e Accendi../ Spegni... dopo 2,5 secondi,
        // ovvero quando la lettura del sensore è terminata

        if(!BTConnesso){
            displaymsg();
            return;
        }
    }
});

```

```

try
{
    //scrivo sulla serial
    BTSocket.getOutputStream().write("2".getBytes());
}
catch (IOException e)
{
    //Toast.makeText(getApplicationContext(), "Errore", Toast.LENGTH_LONG).show();
    finish();
}
umidita.setText("*");
temperatura.setText("*");
Timer timer2=new Timer();
timer2.schedule(new TimerTask() {
    @Override
    public void run() {
        try
        {
            byte[] buffer = new byte[256];
            int l = BTSocket.getInputStream().read(buffer);
            String s = new String(buffer, 0, l);
            String[] ss = s.split(",");
            umidita.setText(ss[0]+"%");
            temperatura.setText(ss[1]+"°C");
        }
        catch (IOException e)
        {
            Toast.makeText(getApplicationContext(), e.toString(), Toast.LENGTH_LONG).show();
        }
    }
}, 2500);
// ritardo di 2,5 secondi per dare tempo ad Arduino di fare 50 letture e far si che
// i valori letti dal sensore (abbastanza lento) possano stabilizzarsi
});
}

} //chiude onCreate()-----

```

```

@Override
public void onBackPressed() {
    if (BTConnesso)
    {
        try
        {
            BTSocket.getOutputStream().write("0".getBytes());//spengo il Led di Arduino
            BTSocket.close(); //chiudo la connessione
        }
        catch (IOException e)
        {
            Toast.makeText(getApplicationContext(), e.toString(), Toast.LENGTH_LONG).show();
        }
    }
    finish();
    Intent i=new Intent(ControlloArduino.this,ListaDispositivi.class);
    startActivity(i);
}

class Connetti extends AsyncTask<Void, Void, Void>
{
    boolean Connessioneok=true;
    String errore;
    @Override
    protected void onPreExecute() {
        super.onPreExecute();
    }
    //
    @Override
    protected Void doInBackground(Void... voids) {
        try
        {
            if (BTSocket== null || !BTConnesso)
            {
                BTAdapter = BluetoothAdapter.getDefaultAdapter();//get the mobile bluetooth device
                BTdevice = BTAdapter.getRemoteDevice(indirizzo);
            }
        }
    }
}

```

```

        BTSocket = BTdevice.createInsecureRfcommSocketToServiceRecord(U);
        BluetoothAdapter.getDefaultAdapter().cancelDiscovery();
        BTSocket.connect();
        Connessioneok = true;
    }
}
catch (IOException e)
{
    Connessioneok = false;
    errore=e.toString();
}
return null;
}

```

```

@Override
protected void onPostExecute(Void aVoid) {
    super.onPostExecute(aVoid);
    if (!Connessioneok)
    {
        //Toast.makeText(getApplicationContext(), errore, Toast.LENGTH_LONG).show();
        BTConnesso = false;
    }
    else
    {
        //Toast.makeText(getApplicationContext(), "OK", Toast.LENGTH_LONG).show();
        BTConnesso = true;
    }
}
}
}

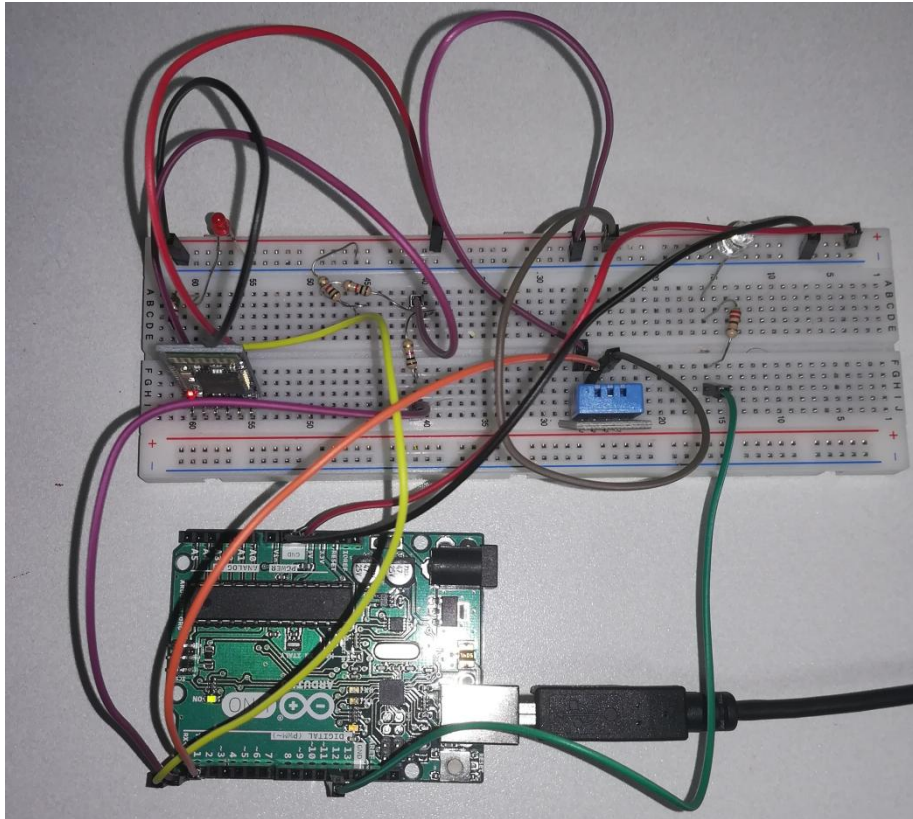
```

```
private void displaymsg()
{
    AlertDialog ad=new AlertDialog.Builder(ControlloArduino.this).create();
    ad.setTitle("MDB 2021 - Attenzione!");
    String msg="Il dispositivo non è ancora pronto oppure non hai selezionato il dispositivo bluetooth HC05";
    ad.setMessage(msg);
    ad.setButton(AlertDialog.BUTTON_POSITIVE, "OK", new DialogInterface.OnClickListener() {
        @Override
        public void onClick(DialogInterface dialog, int which) {
            dialog.cancel();
        }
    });
    ad.show();
}

} //chiude class ControlloArduino-----
```

## Test del progetto

L'immagine, relativa al circuito che ho utilizzato per testare il progetto, fa fedele riferimento allo schema elettrico e ai collegamenti di pagina 2 e pagina 3 di questo tutorial. L'app Android, installata su uno smartphone con schermo da 5.1" e SO Android 8.0.0, non è stata testata con SO più vecchi.



Lo sketch è stato realizzato e caricato su Arduino utilizzando l'IDE Arduino Release 1.8.13



Una volta realizzato il circuito di Arduino e installata l'app Android sullo smartphone, occorre accoppiare lo smartphone con il modulo Bluetooth HC05.

- tenendo acceso il circuito di Arduino, ricerchiamo sul nostro smartphone i dispositivi bluetooth (figura 1)
- quando tra i dispositivi disponibili ci viene proposto il modulo HC05, lo associamo al nostro smartphone inserendo la password di default 1234 (figura 2)

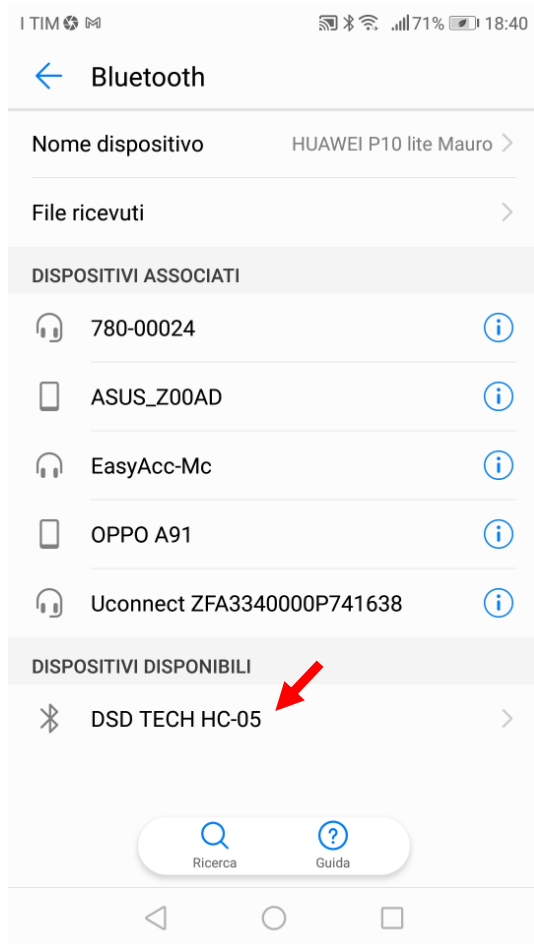


figura 1

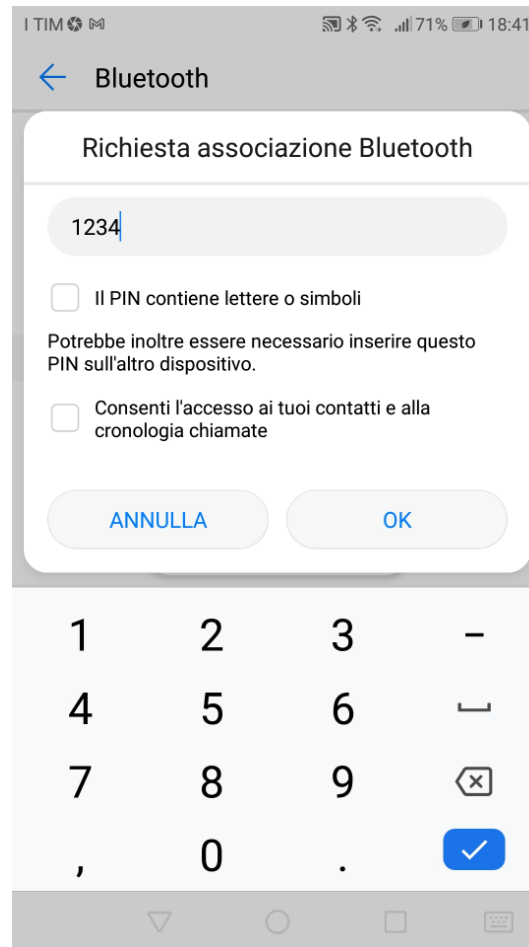


figura 2

A questo punto mandiamo in esecuzione l'app e vediamo come funziona attraverso alcune immagini e un video

Dopo circa due secondi l'attività di avvio (figura 3) richiama l'attività ListaDispositivi: se Bluetooth non è attivo, una finestra di dialogo chiede di attivarlo (figura 4).



figura 3



figura 4

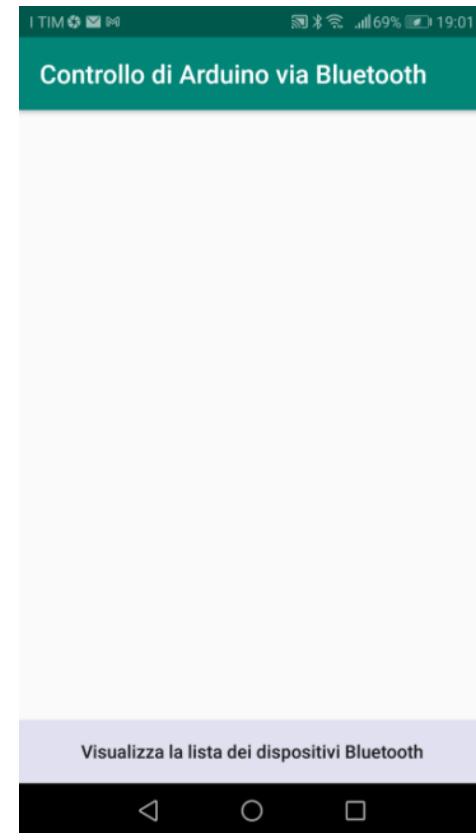


figura 5

Tramite il bottone in basso "Visualizza..." (figura 5), accediamo alla lista dei dispositivi Bluetooth associati al nostro smartphone e selezioniamo il modulo HC05 (figura 6).

Dopo pochi secondi il modulo HC05 si connette allo smartphone, il led collegato al pin STATE del modulo si accende e possiamo controllare il circuito con il nostro smartphone (figura 7).



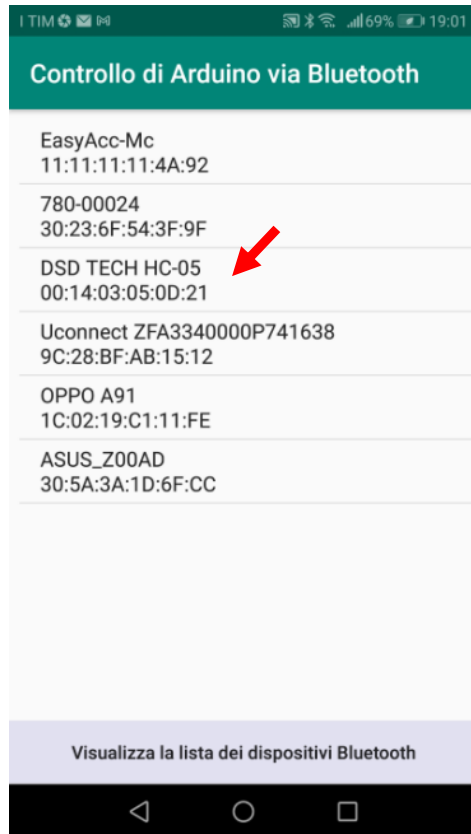


figura 6



figura 7

- Se iniziamo a controllare Arduino prima che il led si accende, l'app ci segnala un errore (figura 8). Lo stesso errore viene segnalato se abbiamo selezionato un dispositivo diverso dal modulo HC05. Nel primo caso attendiamo che si accenda il led su STATE e ripetiamo il controllo, nel secondo torniamo alla schermata precedente e selezioniamo il dispositivo giusto
- Le operazioni di accensione e spegnimento del led sono immediate mentre la lettura dei valori di umidità e temperatura avviene in circa tre secondi, necessari perché i valori letti dal sensore, che non è velocissimo, si stabilizzino. A tal proposito, perché la misura dell'umidità e della temperatura si stabilizzi ulteriormente, consiglio di effettuare, una subito dopo l'altra, 5-6 letture. Durante la lettura del sensore i due bottoni di lettura e di accensione/spegnimento del led sono disabilitati

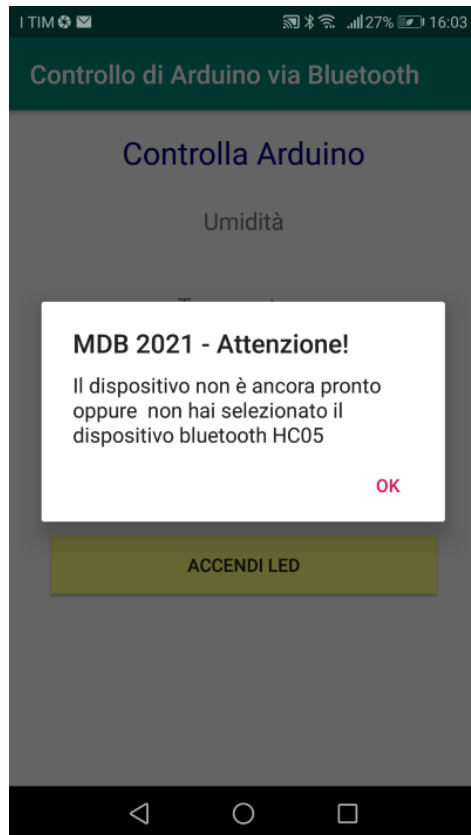


figura 8

- La prova del progetto è illustrata nel video al seguente link: [https://www.youtube.com/watch?v=nY\\_hYWBjxTA](https://www.youtube.com/watch?v=nY_hYWBjxTA)
- [Scarica l'APK dell'app Android e lo sketch di Arduino](#)